

Índice

1.	ESTADO DE DESENVOLVIMENTO	4
2.	ARQUITECTURA GLOBAL.....	5
3.	IMPLEMENTAÇÃO.....	6
4.	UTILIZAÇÃO.....	7

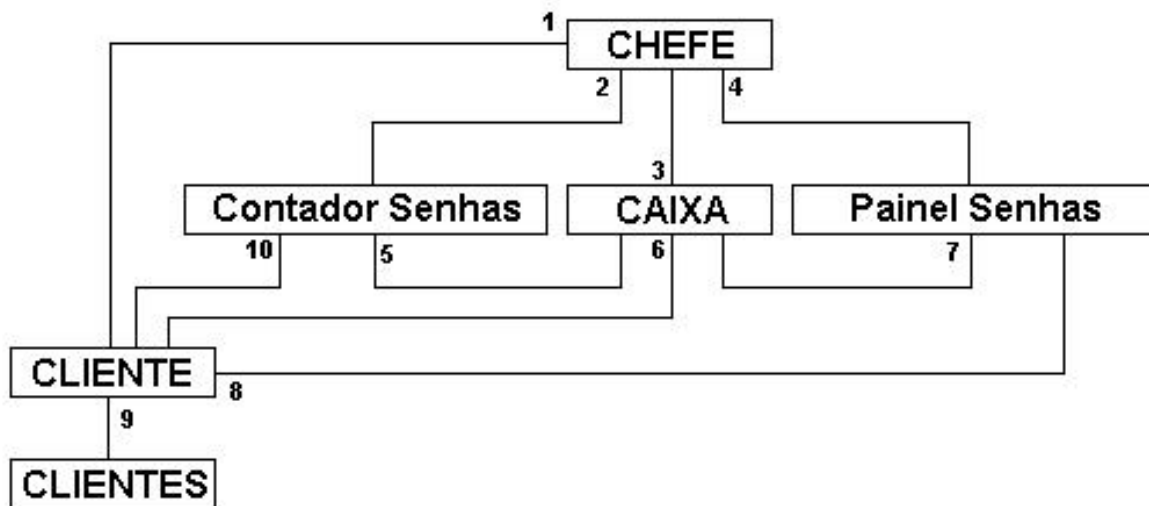
1. Estado de Desenvolvimento

De uma maneira geral o nosso programa funciona bem, tendo em conta os testes por nós efectuados e o facto de as implementações pedidas serem realizadas quase na sua totalidade.

Uma das coisas que o nosso programa ainda não implementa correctamente é o último passo do processo **chefe**, a ordenação cronológica de todas as tarefas que se encontram nos ficheiros “caixaPID.log” criados pelos processos **caixa** e a consequente escrita no ficheiro “work.log”. O não funcionamento desta parte do trabalho deve-se ao facto de não termos conseguido arranjar uma estrutura que se revela-se eficiente para guardar todas as linhas desses ficheiros para uma posterior ordenação, visto que na sua implementação encontramos muitos problemas relacionados com endereçamento de memória. No entanto optámos por implementar esta operação de uma maneira diferente, sendo cada processo **caixa** responsável por escrever cada tarefa por si executada, ao mesmo tempo nos ficheiros “caixaPID.log” e “work.log”, ficando assim todas as tarefas executadas pelas diferentes caixas automaticamente ordenadas cronologicamente no último ficheiro.

É importante referir que um dos aspectos que se dificultou mais o desenvolvimento deste projecto foi o problema de, em determinadas execuções do mesmo programa sob as mesmas condições, ocorreram erros de execução alternadamente, o que atrasou imenso a implementação do trabalho.

2. Arquitectura Global



Ligação	Descrição
1	<i>Thread</i> lançada pelo chefe detecta a chegada de um cliente
2	Chefe cria a região de memória Contador Senhas Chefe incrementa o Contador Senhas quando se verifica 1
3	Chefe lança todos os processos caixa
4	Chefe cria a região de memória Painel Senhas
5	Caixa verifica se há mais clientes para atender através do Contador Senhas
6	Caixa comunica com o cliente e vice-versa para o envio de pedidos e consequente resposta
7	Caixa actualiza o Painel Senhas
8	Cliente olha para o Painel Senhas para ver se chega a sua vez
9	Clientes lança todos os processos cliente
10	Cliente retira uma senha do Contador Senhas quando chega ao balcão

3. Implementação

Para a resolução dos diversos problemas encontrados durante o desenvolvimento do projecto encontramos as soluções respectivas:

Problema	Solução
Indicar à caixa que o tempo de serviço terminou	Caixa lança uma <i>Thread</i> no início que fica responsável por detectar o fim do tempo de serviço e coloca a variável <i>service_end</i> a <i>true</i>
Garantir que o painel fica visível algum tempo	Caixa faz um <i>sleep</i> dum tempo definido como parâmetro após a actualização do painel
Indicar aos clientes que o tempo de serviço terminou	Quando o cliente tenta obter uma chave para a região de memória do contador de senhas ocorre um erro, logo o cliente sabe que o tempo de serviço já terminou
Assinalar fim de pedidos por parte do cliente	Na troca de mensagens entre o cliente e o caixa , após o envio de todas as operações por parte do primeiro, este envia depois uma mensagem com o conteúdo “END” que é lida pela caixa e que sabe que deve parar de receber pedidos

4. Utilização

Para iniciar a execução do nosso programa basta escrever na *shell* “**chefe** *num_caixas tempo_servico*”, e que *num_caixas* é o número de caixas a lançar pelo chefe e *tempo_servico* é a duração de execução do balcão de atendimento.

Depois da execução do **chefe** e do lançamento dos processos **caixa** por parte deste, existem duas hipóteses:

- O utilizador lança clientes manualmente através do comando “**cliente** *num_operacoes tempo_espera*”, em que *num_operacoes* é o número de operações a pedir pelo **cliente** e o *tempo_espera* é o tempo máximo que o **cliente** pode esperar até ser atendido
- O programa lança clientes automaticamente através de “**clientes** *num_clientes intervalo_tempo*”, em que *num_clientes* em que o número de clientes a serem lançados e o *intervalo_tempo* é o tempo médio que o processo demora a lançar cada cliente

O **chefe** termina a sua execução e todos os ficheiros “*.log” ficam guardados no directório onde o processo foi lançado e as regiões de memória e semáforos por ele criados são libertados do sistema.